

REMARKS

Applicants appreciate the thoroughness with which the Examiner has examined the above-identified application. Reconsideration is requested in view of the amendments above and the remarks below.

Drawing objections

The Examiner has objected to the drawings on the grounds that step 28 of Fig. 1 recites the term "limit," a word not used in the specification and claims. Instead, the specification and claims use the term "threshold." The have amended the drawings by replacing the term "limit" with the term "threshold" to correspond the drawings to the specification and claims. A new set of formal drawings including a revised Fig. 1 and a marked-in-red Fig. 1 showing the changes are attached for the Examiner's approval.

Claim objections

Rejection of Claim 19 under 37 CFR 1.75(c)

The Examiner has rejected claim 19 under 37 CFR 1.75(c) on the grounds that it fails to further limit the subject matter of a previous claim. Claim 19 is dependent upon claim 17 which states that "the software update query includes software publisher information identifying a publisher of software to be updated." The applicants have amended claim 19 to state that "the software update query includes software publisher information identifying at least one additional publisher of software to be updated."

By reciting "a publisher" in claim 17, the conventional open ended format of preceding the element by "a" does not exclude additional elements. Claim 19 then narrows the claim by excluding a single publisher and requires that there be at least the original publisher from claim 17 and the "additional publisher." It is believed that this revision to the wording of claim 19 clarifies the claim and places that claim in an acceptable form without changing its scope.

Rejection of Claims 1-8, 12-13 20, and 22-26 under 35 USC §102(b)

Claims 1-8, 12-13 20, and 22-26 have been rejected under 35 USC §102(b) in view of Cole et al. Applicant respectfully traverses this rejection. Cole is directed to a closely related problem of downloading updates that are "consistent" with the computer - "The server computer identifies code updates which are consistent with basic system characteristics of the client computer" column 1 line 47).

The process implemented by Cole et al starts when the client computer contacts the server. This contact is manually initiated by a user who "selects an icon to invoke update manager 32" (column 3 line 15). The server supplies an address for a "basic system information recognizer program" which the client downloads and executes to determine basic system characteristics. These basic characteristics are things that "are not likely to change often such as type of operating system" (column 3 line 49).

The "basic system information recognizer program" sends the information about these basic unchanging characteristics to the server, which then "eliminat[es] the vast majority of the code updates stored" on the basis that they are not "consistent" with the basic characteristics of the client. (column 3 line 56).

Once the possible code updates are separated into "consistent" or "inconsistent" (column 3 line 59), the server sends the client the address of multiple recognizer programs corresponding to the "consistent" updates. It does not send the address of recognizer programs that correspond to "inconsistent" updates. The "consistent" update recognizer programs are fetched and executed to determine if the corresponding update is needed. At this point, the recognizer program may determine that the update has already been installed and is not required (column 4 line 57 - "the client may already have the latest version").

"Next, the client 14 executes each recognizer program" (column 4 line 59) to determine which programs need to be updated. Each recognizer program can assign a "critical" level to an update depending on the results of its analysis (column 5 line 60). The recognizer programs send their results to the server. "[T]he server determines the level of criticality of the respective code updates and builds a selection form for display at the client" (column 6 line 2).

The user manually selects the updates desired from the server displayed "selection form," by selecting from 1) a minimum level of "critical" updates, 2) a maximum level of all updates that are not already installed and are "consistent" with the basic unchanging characteristics of the client or by selecting from a list of all available updates (column 6 line 9). After the user makes the selection, the updates are downloaded and installed.

This process, while related to the problem solved by the present invention differs significantly in several respects from the present invention as disclosed and claimed.

In claim 1 line 9 it specifies that there is “stored user preference information” and that this “information” is “accessed” as part of the update process. In Cole et al. the user manually selects each time from a “displayed” “selection form” to express his preference. Accordingly, Cole et al. does not teach or suggest the “stored user preference information” of claim 1.

In claim 1 line 10 it specifies that there is a determination as to whether the “software update should be automatically downloaded and installed” and that this “determination” is based upon “the user preference information and the evaluated criticality of the software update.”

This “automatic” download and installation of claim 1 contrasts with the non-automatic manual selection process of Coles et al which uses a “displayed” selection form. More importantly, the present invention teaches, and claim 1 claims, that this “automatic” download and install process is possible by making a determination based upon two factors, namely, a “user preference” that is “stored” so that it can be accessed and used when needed, and an “evaluated criticality of the software update” from the criticality check program.

Coles et al. fails to teach or suggest storing the user’s preference and using the same to automatically download and install an update. Accordingly, it is respectfully requested that the rejection under 35 USC § 102(b) be withdrawn.

In claim 4, the invention is further defined as starting with a “criticality rating” in the software update information and then “adjusting the criticality rating from the software

update information" to produce the "evaluated criticality" and "comparing" that to a "user criticality threshold" that is stored in the "user preference information."

Again, Coles et al fails to teach this threshold and comparison step of the present method that produces an "automatic" download. The rejection under 35 USC § 102(b) should be withdrawn.

Rejection under 35 USC § 103

Claims 14-16 and 21 stand rejected under 35 USC § 103 as being obvious from Cole et al. in view of the prior art. Applicant respectfully traverses this rejection.

With respect to claims 14 and 15, the applicants agree that automated download and install of updates is a known method of the art, but this known method has heretofore proven to be unsatisfactory and the present invention is designed specifically to address the shortcomings of that automated process. For that matter, Cole et al. is also designed to address the problems of the prior art automated process - "some clients may not wish to obtain the update" (column 1 line 35).

The solution taught by Cole et al. is to **abandon** the prior art automated download and install process and require a manually initiated update (user selects the "icon") and a manual selection of desired updates from a "selection form." Cole attempts to make it easier on the user to perform this manual step by reducing the number of choices. First he eliminates "inconsistent" updates as determined by the "basic system information recognizer program," which are not suitable for the computer's unchanging operating system or BIOS, etc. Second, he eliminates unnecessary updates - already installed as determined by the update recognizer programs. Third, he identifies some updates as

critical, and finally, he provides the user with some bulk choice options allowing the user to select all “consistent” updates (maximum) or all critical updates (minimum).

The solution taught by the present invention is the opposite of Cole et al. The present invention teaches that the advantages of the prior art automatic download and install process should be **retained**, not abandoned. This is achieved by the claimed method of storing the user’s preferences and using the computer, not the user, to make the choice between updates, guided by a “stored” user preference.

The prior art method of automatically initiated updates and automatic download and install cannot be combined with Cole et al when Cole teaches directly away from the prior art. How can one automatically initiate the update process and automatically download and install updates if the user is not there to manually select the desired updates as required by Cole? Clearly Cole knew of the prior art automatic process, but decided to abandon it. The Examiner cannot use hindsight to apply what Cole decided to discard.

Claims 17 and 19 have been rejected under Cole in view of Pedrizetti et al. The applicants vigorously deny that there is any teaching in Pedrizetti et al. or Cole of identifying one or more specific publishers in the claimed “software update query.” Both Pedrizetti et al. and Cole et al. disclose a manual update and selection process, not the claimed automatic process. Neither discloses identifying one or more specific publishers in an update query as set forth in claims 17-19. Pedrizetti et al. sends an optional vendor specific DLL, when an update exists for that vendor’s software, but this occurs after the software update query has been sent by the client and after it has been determined that there is an update for software by that vendor. There is no teaching in Pedrizetti et al. that

the "software update query" itself should identify one or more publisher/vendors as in the present invention.

It is respectfully submitted that the application has now been brought into a condition where allowance of the entire case is proper. Reconsideration and issuance of a notice of allowance are respectfully solicited.

Respectfully submitted,



Peter W. Peterson
Reg. No. 31,867

DeLIO & PETERSON, LLC
121 Whitney Avenue
New Haven, CT 06510-1241
(203) 787-0595

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date indicated below as first class mail in an envelope addressed to the Mail Stop NON-FEE AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 2231301450.

Name: Carol M. Thomas Date: October 8, 2004 Signature: Carol Thomas
ibmf100335000amdA

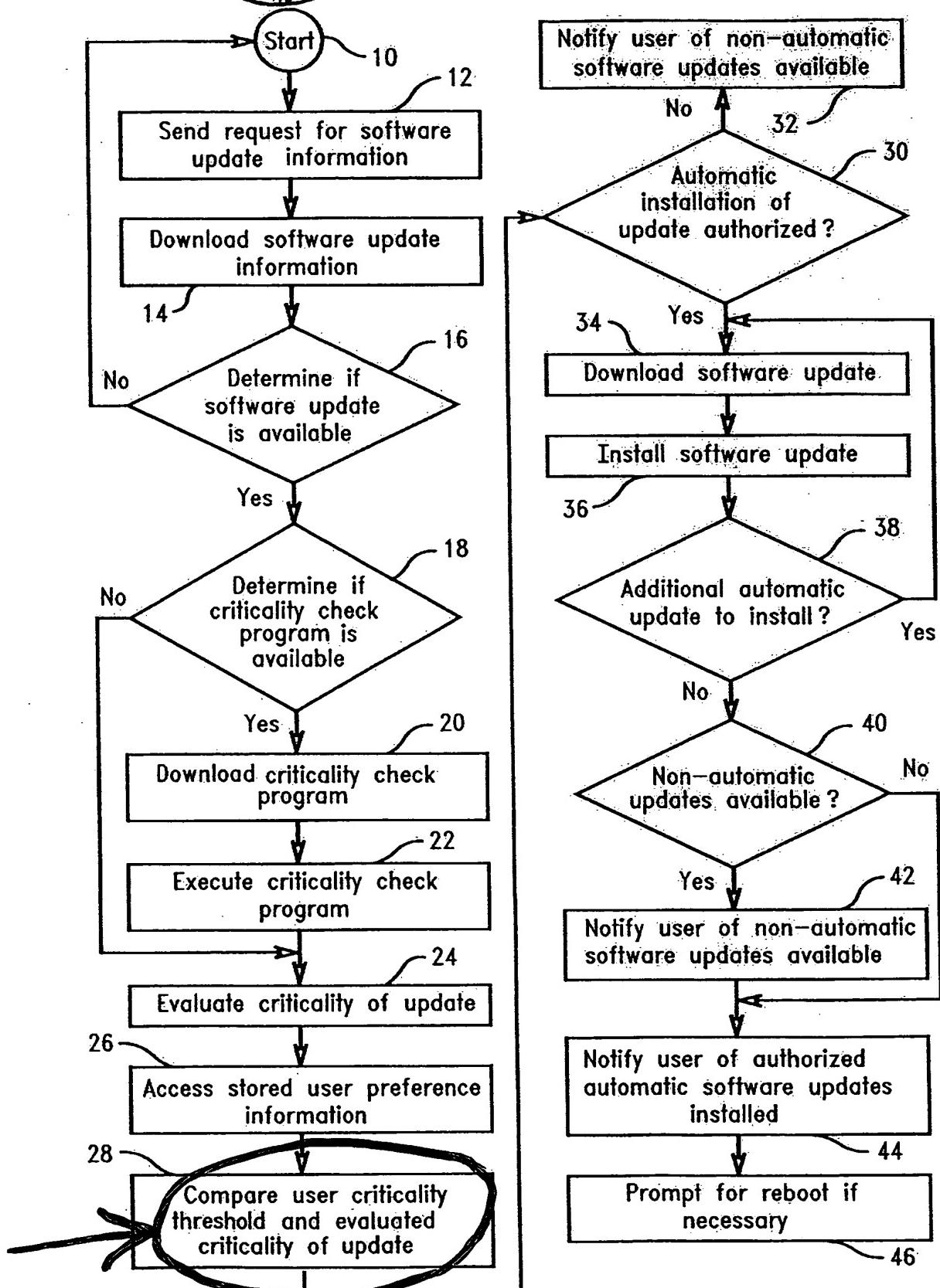


FIG. 1